

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan membahas tentang teori-teori pendukung penelitian mengenai prediksi, intensitas curah hujan, Jaringan Saraf Tiruan, dan Backpropagation.

2.1 Kajian Pustaka

Penelitian mengenai sistem prediksi telah banyak dilakukan. Nur Endah Sari dan Edi Sukirman (2011) melakukan penelitian untuk memprediksi cuaca yang digunakan untuk rekomendasi penerbangan di Bandar Udara Raja Haji Fisabilillah. Penelitian tersebut menggunakan metode berbasis logika *fuzzy* dengan menggunakan nilai masukan berupa data suhu udara, tekanan udara, kelembaban udara dan kecepatan angin. Setiap data masukan dibagi menjadi 3 *cluster* yang berbeda seperti untuk variabel suhu dibagi menjadi *cluster* rendah, sedang, dan tinggi. Pada penelitian tersebut menggunakan 365 data yang digunakan untuk membangun logika, yaitu data dari bulan Maret 2010 hingga Februari 2011. Hasil prediksi dari penelitian tersebut yaitu dari 122 hari yang diuji, terdapat 22 hari untuk kondisi hujan yang tidak sesuai dengan data aktual sehingga menghasilkan keakuratan sebesar 61,73% .

Penelitian lain oleh Candra Dewi, Dany Primanita Kartikasari, Yusi Tyroni Mursityo (2014) yang melakukan prediksi cuaca menghasilkan akurasi diatas 90%. Data yang digunakan sebagai masukan dalam melakukan prediksi yaitu berupa data harian suhu, kelembaban udara, kecepatan angin dan tekanan udara yang didapatkan dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) dari Kecamatan Karangploso, Kabupaten Malang, dari bulan Januari 2011 sampai dengan Mei 2012 . Berdasarkan pengujian diketahui bahwa kemampuan ANFIS dalam melakukan peramalan menghasilkan akurasi yang bagus apabila dilakukan pada data yang berada pada musim yang sama namun akurasi akan menurun jika data pada musim yang berbeda. Akurasi tertinggi pada proses prediksi curah hujan tersebut adalah 76,93%. Lalu jika prediksi dilakukan dengan memisah data pada musim yang berbeda akan menghasilkan akurasi 85,16% untuk musim kemarau dan 87,10% untuk musim penghujan.

Indrabayu , Nadjamuddin Harun, M. Saleh Pallu, Andani Achmad, dan Febi Febriyati (2012) pada penelitian mengenai prediksi curah hujan dengan menggunakan metode *Radial Basis Function Neural Network*. Penelitian tersebut menggunakan data masukan berupa curah hujan, kecepatan angin, kelembaban, tekanan, dan temperatur yang didapat dari BMKG Wilayah IV Makassar. Data dikelompokkan dari tahun 2004 hingga 2008 dijadikan sebagai data masukan dan data pada tahun 2009 dijadikan sebagai target pelatihan. Penelitian ini menghasilkan keakuratan prediksi tahun 2010 sebesar 69,37% dengan RMSE sebesar 0,2549.

Daneswara Jauhari, Alfian Himawan, dan Candra Dewi (2016) melakukan sebuah penelitian untuk memprediksi distribusi air PDAM pada kota Malang,

sistem ini membuat sebuah prediksi produksi jumlah air untuk mengatasi kekurangan maupun kelebihan jumlah produksi jumlah air dengan menggunakan data pemakaian air di bulan-bulan sebelumnya. Data yang digunakan pada penelitian ini yaitu data rentet waktu atau *time series* berupa data penggunaan air pada bulan-bulan sebelumnya. Akurasi tertinggi dari penelitian ini adalah 97,99% dengan parameter terbaik adalah *learning rate* 0.1, data latih berjumlah 80 serta data uji berjumlah 11, jumlah node pada hidden layer 5, dan jumlah perulangan 2000.

Metode yang sama juga dipakai oleh Siti Amiroch (2015) pada penelitiannya yang berjudul "Prediksi Harga Saham Menggunakan Jaringan Saraf Tiruan Backpropagation". Data masukan yang digunakan pada sistem yaitu data harga saham penutupan yang diberlakukan secara *time series* serta data keluaran berupa prediksi harga saham pada hari berikutnya. Dari hasil uji dari sistem ini, diperoleh nilai prediksi harga saham dengan selisih yang kecil dengan data sebenarnya sehingga dapat disimpulkan bahwa sistem ini memiliki nilai akurasi yang cukup tinggi.

Berdasarkan penelitian yang telah dijabarkan diatas, penulis akan menggunakan metode jaringan saraf tiruan Backpropagation. Metode jaringan saraf tiruan Backpropagation sering digunakan sebagai metode prediksi pada data time series dan kasus non-linear pada banyak permasalahan. Metode Backpropagation ini juga memiliki ketangguhan dalam menganalisa pola pada data yang besar. Sehingga dapat disimpulkan bahwa metode ini lebih *general* dan fleksibel untuk melakukan prediksi pada data time series seperti data intensitas curah hujan.

2.2 Dasar Teori

2.2.1 Intensitas Curah Hujan

Hujan adalah fenomena turunnya air dari langit ke permukaan bumi karena proses kondensasi uap air di atmosfer menjadi butir-butir air yang cukup berat untuk jatuh ke permukaan bumi. Dua proses yang terjadi bersamaan dapat mendorong udara semakin jenuh, yaitu pendinginan udara atau penambahan uap air ke udara (Anjayani & Haryanto, 2009).

Intensitas curah hujan adalah jumlah air yang jatuh ke permukaan bumi selama periode waktu tertentu. Intensitas curah hujan dinyatakan dengan satuan mm (milimeter) per satuan waktu tertentu. Berikut adalah derajat dan intensitas curah hujan bulanan (Anjayani & Haryanto, 2009).

Tabel 2.1 Derajat dan intensitas curah hujan bulanan

Derajat Hujan	Intensitas Curah Hujan (mm/bulan)
Intensitas Curah Hujan Bulanan Rendah	0-100

Intensitas Curah Hujan Bulanan Menengah	>100-300
Intensitas Curah Hujan Bulanan Tinggi	>300-500
Intensitas Curah Hujan Bulanan Sangat Tinggi	>500

Tabel 2.1 menggunakan satuan waktu per bulan. Dari pemaparan tabel derajat dan intensitas curah hujan, dapat diketahui bahwa curah hujan dengan takaran dibawah 0 sampai dengan 100 mm per bulan termasuk intensitas curah hujan rendah, dengan takaran antara lebih dari 100 hingga 300 termasuk intensitas curah hujan menengah, dan seterusnya.

2.2.2 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) adalah metode pemrosesan informasi untuk menyelesaikan masalah-masalah yang rumit dengan mempelajari pola pola tertentu dalam kumpulan data. Jaringan Saraf Tiruan memproses data statistik non-linier dengan cara mengadopsi cara kerja jaringan saraf pada otak manusia untuk memproses informasi yang didapat (Siang, 2009).

Terdapat 3 elemen dasar pada jaringan saraf tiruan, yaitu bobot (*weight*), nilai ambang (*threshold*), dan fungsi aktivasi. Setiap *input* yang datang selalu memiliki bobot masing-masing. Bobot yang masuk tersebut dapat berubah sesuai nilai ambang dan mendapat aktivasi dari sel saraf yang kemudian dari aktivasi tersebut dapat menjadi fungsi aktivasi yang menghasilkan output pada sel saraf.

Jaringan saraf tiruan memiliki 3 lapisan utama, yaitu:

1. Lapisan Masukan (*Input layer*)

Lapisan ini berfungsi menerima input dari jaringan saraf luar. Aktivasi pada lapisan ini yang akan digunakan pada jaringan saraf tiruan yang kemudian akan disalurkan ke lapisan tersembunyi atau lapisan luaran.

2. Lapisan Tersembunyi (*hidden layer*)

Pada lapisan ini, aktivasi dari lapisan ini ditentukan oleh bobot yang masuk serta aktivasi unit masukan. Pada lapisan ini dapat terjadi perubahan bobot untuk menyesuaikan dengan target yang akan dicapai.

3. Lapisan Luaran (*Output layer*)

Lapisan ini berfungsi untuk menghitung nilai keluaran. Lapisan ini menghitung nilai keluaran dengan menerapkan fungsi aktivasi pada lapisan luarannya.

2.2.3 Normalisasi dan Denormalisasi Data

Pada jaringan saraf tiruan, *input* yang digunakan merupakan data dengan rentang antara 0 hingga 1 untuk fungsi biner dan -1 hingga 1 untuk fungsi bipolar. Maka diperlukan sebuah metode untuk melakukan pengolahan data guna merubah jangkauan data seperti yang dispesifikasikan diatas yang disebut dengan normalisasi, sedangkan denormalisasi merupakan sebaliknya, yaitu merubah data

dengan jangkauan tertentu ke jangkauan aslinya. Data yang digunakan dapat di skalakan dalam range 0 hingga 1, namun mengingat fungsi *asimtotik* yang nilainya tidak pernah mencapai 0 maupun 1, maka rentang yang digunakan dalam penelitian ini yaitu antara 0,1 hingga 0,9. Berikut adalah rumus metode normalisasi dan denormalisasi:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \cdot 0,8 + 0,1 \quad (2.1)$$

$$x = \frac{x' - 0,1}{0,8} (x_{max} - x_{min}) + x_{min} \quad (2.2)$$

Keterangan:

- x' = data hasil normalisasi
- x = data asli / hasil denormalisasi
- x_{min} = data dengan nilai minimum dari semua data
- x_{max} = data dengan nilai maksimum dari semua data

2.2.4 Backpropagation

Algoritme *Backpropagation* atau bisa disebut dengan Algoritme Perambatan Mundur merupakan algoritme yang menerapkan proses pembelajaran secara terarah atau *supervised learning* pada jaringan saraf tiruan untuk menemukan bobot pada setiap neuron yang sesuai dengan target melalui data pembelajaran (*pelatihan data*) (Siang, 2009). Algoritme ini merupakan salah satu metode yang cukup populer untuk digunakan dalam bidang *forecasting*. Arsitektur jaringan saraf tiruan *backpropagation* akan ditunjukkan pada Gambar 2.1.

Menurut Siang (2009), algoritme ini memiliki 3 fase pelatihan, yaitu :

1. Propagasi Maju

Pada fase ini, setiap sinyal input diteruskan ke lapisan tersembunyi sampai lapisan luaran dengan menggunakan fungsi aktivasi yang telah ditentukan sebelumnya. Fase ini menghasilkan output aktivasi propagasi.

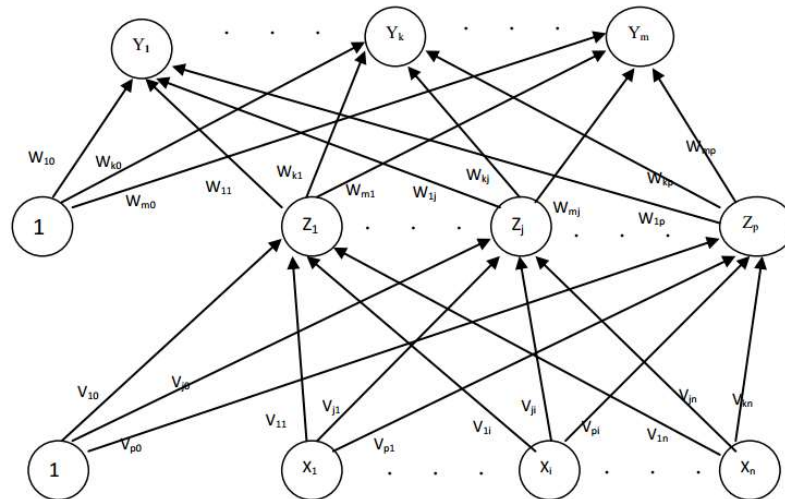
2. Propagasi Mundur

Pada fase ini, selisih antara keluaran dengan target yang diinginkan (kesalahan) yang terjadi di propagasi mundur dan dilakukan perubahan bobot guna menghasilkan keluaran yang mendekati dengan target.

3. Perubahan bobot

Pada fase ini, dilakukan modifikasi bobot untuk mengurangi kesalahan yang terjadi.

Ketiga fase tersebut dilakukan secara berulang hingga mencapai kondisi berhenti (*stop condition*) .



Gambar 2.1 Arsitektur Jaringan Saraf Tiruan *Backpropagation*

2.2.4.1 Fungsi Aktivasi *Backpropagation*

Fungsi aktivasi yang akan digunakan oleh jaringan saraf tiruan *Backpropagation* memiliki beberapa syarat yang harus dipenuhi, diantaranya adalah kontinu, terdiferensial dengan mudah, serta merupakan fungsi yang tidak turun. Fungsi yang memenuhi persyaratan tersebut adalah fungsi aktivasi sigmoid biner. Berikut merupakan persamaan fungsi *sigmoid biner* yang ditunjukkan oleh persamaan 2.3.

$$f(x) = \frac{1}{1 + e^{-1}} \quad (2.3)$$

2.2.4.2 Langkah - langkah *Backpropagation*

Langkah 0 :

Inisialisasi bobot dengan bilangan acak

Menentukan maksimum epoch, target error, dan *learning rate*

Inisialisasi epoch = 0

Langkah 1:

Jika epoch < maksimum_epoch dan MSE < target_error , maka kerjakan langkah 2-9

Langkah 2:

Untuk setiap data pelatihan lakukan langkah 3-8

Fase Propagasi Maju :

Langkah 3:

Tiap unit masukan menerima sinyal input dan meneruskannya ke jaringan tersembunyi diatasnya

Langkah 4:

Hitung semua keluaran di hidden layer ($z_j=1,2,3,\dots,p$)

$$z_in_j = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (2.4)$$

$$z_j = f(z_in_j) = \frac{1}{1 + e^{-z_in_j}} \quad (2.5)$$

Keterangan:

z_in_j =sinyal masukan pada hidden layer j

v_{j0} =Bias hidden layer ke $-j$

v_{ij} =bobot antara untuk input layer ke i dan hidden layer ke j

x_i =unit input layer ke- i

z_j =unit input ke layer- j

i =urutan unit *input layer*

j =urutan unit *hidden layer*

p =jumlah maksimum unit pada *hidden layer*

Langkah 5:

Hitung semua keluaran jaringan unit y_k ($k = 1,2,3, \dots$)

$$y_in_k = w_{0k} + \sum_{j=1}^n z_j w_{jk} \quad (2.6)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_in_k}} \quad (2.7)$$

Keterangan:

y_in_k =sinyal masukan output ke $-k$

w_{0k} =bias *hidden layer* ke $-k$ (untuk bobot awalnya, nilainya random antara -0,5 dan 0,5)

w_{jk} =bobot antara *hidden layer* ke $-j$ dan output ke $-k$ (untuk bobot awal, nilainya *random* antara -0,5 dan 0,5)

y_k =aktivasi nilai *output layer* ke $-k$

n =jumlah unit maksimum pada *output layer*

Fase Propagasi Mundur:

Langkah 6:

Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1,2, \dots, m$)

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k) \quad (2.8)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot *layer* dibawahnya (langkah 7).

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\Delta w_{jk} = \alpha \delta_k z_j ; \quad k = 1, 2, \dots, m ; \quad j = 0, 1, \dots, p ; \quad (2.9)$$

Hitung korelasi bias Δw_{k0} yang nantinya digunakan untuk memperbarui nilai w_{0k} .

$$\Delta w_{0k} = \alpha \delta_k \quad (2.10)$$

Keterangan:

δ_k =faktor koreksi error bobot w_{kj}

t_k =target output ke - k

α =laju percepatan (*learning rate*)

Δw_{jk} =nilai koreksi error bobot w_{kj}

Δw_{0k} =nilai koreksi error bias w_{k0}

z_j =aktivasi hidden layer ke - j

Langkah 7:

Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, m$)

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.11)$$

Faktor koreksi error (δ) unit tersembunyi

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (2.12)$$

Hitung suku perubahan bobot v_{ji} (yang dipakai nanti untuk merubah bobot v_{ji})

$$\Delta v_{ij} = \alpha \delta_j x_i ; \quad j = 1, 2, \dots, p ; \quad i = 0, 1, \dots, p ; \quad (2.13)$$

Dihitung juga koreksi bias Δv_{0j} (yang nantinya akan dipakai untuk merubah bobot Δv_{0j})

$$\Delta v_{0j} = \alpha \delta_j \quad (2.14)$$

Fase Perubahan Bobot:

Langkah 8:

Hitung semua perubahan bobot

Perubahan bobot garis yang menuju unit keluaran :

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.15)$$

$$w_{0k}(lama) = w_{0k}(lama) + \Delta w_{0k} \quad (2.16)$$

Perubahan bobot garis yang menuju unit tersembunyi :

$$v_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji} \quad (2.17)$$

$$v_{0j}(lama) = v_{0j}(lama) + \Delta v_{0j} \quad (2.18)$$

Langkah 9:

Memeriksa semua *stopping condition*

Stop condition digunakan sebagai kondisi untuk menghentikan proses pada fase pelatihan. Ada dua *stop condition* yang dapat digunakan:

1. Dengan membatasi iterasi (epoh) yang dijalankan. Satu iterasi (epoh) adalah perulangan langkah ke – 3 sampai dengan langkah ke – 8 untuk keseluruhan data pelatihan.
2. Dengan membandingkan Mean Average Percentage Error (MAPE) dengan target error. Jika MAPE kurang dari target error maka lanjutkan.

2.2.5 Algoritme Nguyen-Widrow

Algoritme *Nguyen-Widrow* adalah sebuah algoritme yang digunakan untuk mengurangi waktu pelatihan pada jaringan saraf tiruan. Pada penelitian yang dilakukan oleh Mishra, Mittal, Mirja (2014), algoritme ini digunakan untuk melakukan mengkompresi data citra dan berhasil mengurangi waktu pelatihan, dan meningkatkan performa dari jaringan saraf tiruan. Terdapat beberapa langkah dalam melakukan Algoritme Nguyen-Widrow, yaitu sebagai berikut:

Langkah 1:

Inisialisasi bobot awal v_{ij} dengan bilangan acak antara -0.5 hingga 0.5

Langkah 2:

Hitung nilai vektor v_{ij}

$$||v_j|| = \sqrt{v_{1j}^2 + v_{2j}^2 + v_{3j}^2 + \dots + v_{nj}^2} \quad (2.19)$$

Langkah 3:

Hitung nilai faktor skala β yang dipakai untuk menghitung bobot v_{ij}

$$\beta = 0.7(p)^{\frac{1}{n}} = 0.7\sqrt[n]{p} \quad (2.20)$$

Langkah 4:

Hitung bobot v_{ij} yang dipakai sebagai inisialisasi

$$v_{ij} = \frac{\beta v_{ij}(lama)}{||v_j||} \quad (2.21)$$

Keterangan:

β	=faktor skala
v_{ij}	=bobot <i>input layer</i> ke <i>hidden layer</i>
$ v_j $	=vektor bobot ke <i>hidden layer</i>
p	=jumlah unit tersembunyi
n	= jumlah unit masukan

2.2.6 Mean Percentage Absolute Error (MAPE)

MAPE merupakan rata-rata dari jumlah *error* atau galat yang bersifat absolut atau mutlak. MAPE digunakan untuk mengetahui akurasi yang dihasilkan dari proses peramalan, seperti peramalan menggunakan jaringan saraf tiruan *Backpropagation*. Persamaan MAPE akan dijabarkan dalam persamaan (2.22) .

$$MAPE(\%) = \frac{\sum_{i=1}^n |y' - y|}{n} \frac{100}{y} \% \quad (2.22)$$

Keterangan:

y'	=Nilai hasil peramalan (keluaran sistem)
y	=Nilai aktual (target seharusnya)
n	=Banyak data